Micro. Notes. 02

Loop (with Counter)

- We Can loop for ~~specific~~ Specific number of times using DJNZ instruction

DJNZ Counter, TARGET

↳ register OR Port

- This instruction will Decrement the Counter then Check if it ~~is~~ not equal Zero then Jump to target other wise, ~~go~~ Continue normal Flow of execution

```
      MOV   R2, #10     } Loop For 10 times
LOOP: DJNZ  R2, LOOP
```

- 8051 Registers are 8-bit So we Can loop to 255 if we need more we should use neested Loop

Example : Loop for 700 times

```
       MOV   R1, #7
Loop1: MOV   R2, #100
Loop2: DJNZ  R2, Loop2   → inner
       DJNZ  R1, Loop1   → outer
```

Conditional Jump

		Jump
JZ	target	→ if A = 0
JNZ	target	→ if A ≠ 0
DJNZ	R0, target	
CJNE	R0, #80, target	→ if R0 ≠ 80
JC	target	→ if Carry = 1
JNC	target	→ if Carry = 0
JB	P0.1, target	→ if bit(P0.1) = 1
JNB	P1.0, target	→ if P1.0 = 0
JBC	C, target	→ if C = 1

JBC Instruction

- This Instruction Check if the bit = 1
- If that is true Jump and Clear that bit

CJNE instruction (slide 222)

- This instruction is used to Compare two Values
- The Comparision is performed by Subtracting the two Values without Change Values in registers
- It only change the Carry

```
           CJNE  A, R0, Not_EQUAL
           .....            → jump not Carry
NOT_EQUAL: JNC, A_G_R0
```

- If the Carry = 1 that mean [A − R0] → result is negative → A < R0
- If the Carry = 0 A ≥ R0

Un Conditional Jump

1 Short Jump

SJMP target

- It take only 2 Bytes
- the first byte For opcode
- the Second byte is a relative address (−128 → 127)

Address	Op Code	Instruction
0007	XX 06	SJMP target
0009	XX●	NOP
- - - -		
000F		target: NOP

0F − 09 = 06

- لاحظ أنه يتم حساب الـ relative address بالإعتماد على عنوان الأمر التالي إضافة إلى عنوان الذي يتم الإنتقال إليه

~~000F - 0009~~

000F - 0009 = 06

- لابد أن يكون الفرق في حدود -128, 127

- جميع Conditional Jump من نوع Short Jump

[2] Long Jump

- It take 3 bytes
- The first byte for opcode
- The other two bytes is the absolute Address of the target

LJMP target

The CALL Instructions

[1] Long Call (LCALL target)

- 3-bytes instruction
- there are two bytes for address

~~[2] Short Call~~

[2] absolute Call (ACALL target)

- 2-bytes instruction
- 11 bits are used for address

- The CALL instruction Store the Content of Program Counter in Stack before Jump
- The RET instruction is used to pop [PC] from Stack and back to instruction after Call

Address	instruction
0002	LCALL XYZ
0005	Nop

Stack

0A	
09	
08	

SP = 07

after LCALL

0A	
09	00
08	05 → Low First

SP = 09

Look At slide 117

- pop
- push

بترتيب عكس

Delay in 8051

- Delay depends on
  1. Crystal frequency
  2. Design of 8051

For 8051

* Crystal freq. = 11.0592 MHz
* Clocks per Machine Cycle
  - depend on Design
  - Default = 12

To Compute machine Cycle

$$\text{machine Freq.} = \frac{\text{Crystal Freq.}}{\text{Clocks per Machine Cyc.}}$$

$$\text{machine cycle} = \frac{1}{\text{machine Freq.}}$$

Common no. of machine cycles for Common Instruction

MOV	1
DEC	1
INC	1
DJNZ	2
LJMP	2
SJMP	2
NOP → no operation	1
MUL AB	4
Push	2
Pop	2
RET	2

Look At examples of Slides 123 → 127

Example

Write Delay Subroutine to introduce 100 ms delay

$$\text{machine freq.} = \frac{11.0592}{12}$$
$$= 921.6 \text{ KHz}$$
$$\text{machine cycle} = 1/921.6 \text{ KHz}$$
$$= 1.085 \ \mu s$$

The required no. of machine cycles

$$\text{no. cycle} = \frac{100 \text{ ms}}{1.085 \ \mu s}$$
$$= 92165.89$$
$$\simeq 92166 \text{ cycles}$$

We try to use two nested Loop

		Cycle
Delay:	MOV R1, ~~#~~ 200 → X	1
Outer:	MOV R2, ~~#~~ 228 → Y	1
inner:	DJNZ R2, inner	2
	DJNZ R1, outer	2
	RET	2

MOV → 1
MOV → 1 * X
DJNZ → 2 * X * Y
DJNZ → 2 * X
RET → 2

$$92166 = 3 + 3X + 2XY$$

Assume $X = 150$
$Y \simeq 305 > 255$

Assume $X = 200$
$Y \simeq 229$